

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
26 April 2001 (26.04.2001)

PCT

(10) International Publication Number  
**WO 01/29780 A1**

(51) International Patent Classification<sup>7</sup>: G07B 17/00

(21) International Application Number: PCT/US00/41322

(22) International Filing Date: 19 October 2000 (19.10.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/154,523 19 October 1999 (19.10.1999) US

(71) Applicant (for all designated States except US):  
STAMPS.COM [US/US]; Suite 1040, 3420 Ocean  
Park Boulevard, Santa Monica, CA 90405 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): WINSLOW, Richard,  
Baker [US/US]; 4133 Jasmine Avenue, Culver City, CA  
90232 (US).

(74) Agent: REED, Derrick, W.; Christie, Parker & Hale,  
LLP, 350 W. Colorado Blvd., P.O. Box 7068, Pasadena, CA  
91109-7068 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,  
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,  
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,  
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,  
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,  
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,  
TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

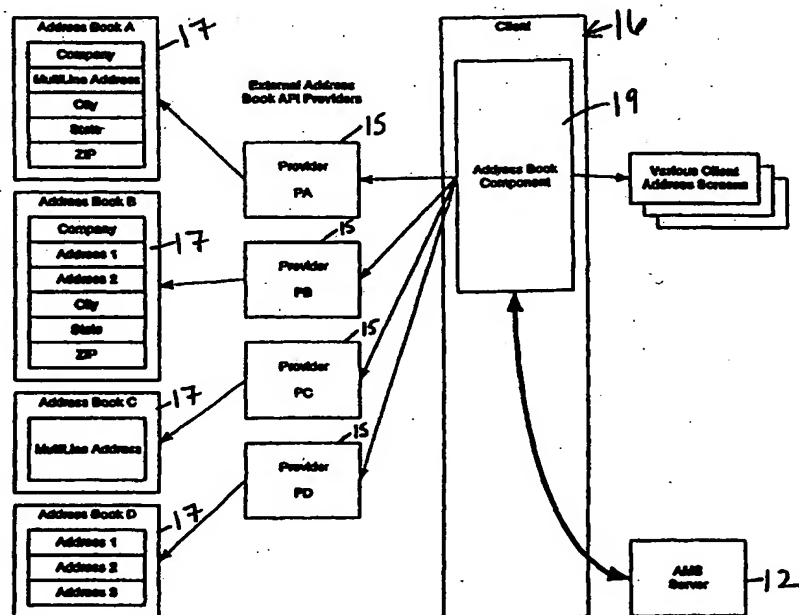
(84) Designated States (*regional*): ARIPO patent (GH, GM,  
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian  
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European  
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,  
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,  
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

- With international search report.
- Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.

[Continued on next page]

(54) Title: ADDRESS MATCHING SYSTEM AND METHOD



(57) Abstract: An address matching system that maintains a central database of valid addresses has been designed. According to the invention, address matching requests are received by the system from a plurality of remote users located at respective machines ("clients"). Those requests are processed at the central system, and the results are returned to the client that made the request. In addition, novel techniques are implemented for improving the likelihood of obtaining one or more matches from a request.

WO 01/29780 A1



---

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## ADDRESS MATCHING SYSTEM AND METHOD

## CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application claims the benefit of the filing date of United States Provisional Patent Application Serial No. 60/154,523, filed October 19, 1999 and entitled "USER ADMINISTRATION FOR ON-LINE POSTAGE SYSTEM", the entire contents of which is hereby expressly incorporated by reference.

## FIELD OF THE INVENTION

The present invention relates to secure printing of value-bearing items (VBI), such as postage for letters and other items to be delivered by the United States Postal Service (USPS). More specifically, the invention relates to a system and method for validating addresses using a central address matching database, with such address matching being conducted over a communications network.

## BACKGROUND OF THE INVENTION

A significant percentage of the United States Postal Service (USPS) revenue is from metered postage. Metered postage is generated by utilizing postage meters that print a special mark, also known as postal indicia, on mail pieces. Generally, printing postage and any other VBI can be carried out by using mechanical meters or computer-based systems. Conventionally, a business or other entity will have a meter at its place of business, and will use the meter to print postal indicia on mail pieces or on labels that are then affixed to the mail pieces.

With respect to computer-based postage processing systems, the USPS, under the Information-Based Indicia Program (IBIP), has published specifications for IBIP postage meters that identify a Postal Security Device (PSD). The PSD, in conjunction with the user's personal computer and printer, functions as the IBIP postage meter. The USPS has published a number of documents describing the PSD specifications, the indicia specifications and other related and relevant information. There are also security standards for printing other types of VBI, such as coupons, tickets, gift certificates, currency, vouchers, and the like.

1 One of the standards required by the USPS is that the  
address of the intended recipient of a piece of mail be verified  
by comparing the entered address with a database of valid  
addresses provided by the USPS. Conventionally, this requires  
5 that the sender have the database installed on their local  
computer and compare the intended destination address with the  
addresses in the database. Moreover, when the USPS provides a  
new release of addresses, each sender is required to obtain the  
updated database to continue to validate addresses. Thus, such  
10 a system can be burdensome to its users.

10 In addition, since addresses vary widely in terms of number  
of elements in the address and various address formats, address  
matching can be a difficult undertaking, even with a current list  
of all valid addresses available to the sender.

15 Therefore, it would be desirable to have a method and system  
for performing address matching from a single, central location,  
thereby doing away with the need for each remote user to maintain  
an updated database of valid addresses. In addition, it would  
be desirable to have such a system and method that allows  
20 addresses to be validated once from a particular sender, such  
that when the sender desires to use a previously validated  
address again, there is no need to perform address matching a  
second time. Moreover, it would be desirable to have a method  
and system that takes a desired address and manipulates the  
25 format into multiple structures before comparing the address with  
a set of valid addresses in order to increase the likelihood of  
obtaining a match. The present invention addresses these  
desirable features.

#### 30 SUMMARY OF THE INVENTION

30 In accordance with one aspect of the present invention, an  
address matching system that maintains a central database of  
valid addresses has been designed. According to the invention,  
address matching requests are received by the system from a  
plurality of remote users via respective devices ("clients").  
35 Those requests are processed at the central system, and the  
results are returned to the client that made the request.

In one embodiment, the central system, in cooperation with  
the client, utilizes novel address matching techniques to

1 increase the likelihood of finding a match for a received  
address. Preferably, the invention utilizes one or more query  
permuters to arrange the address data in different formats in an  
effort to increase the likelihood of finding a match for a  
particular address.

5 In another aspect, the invention describes a method for  
matching an address entered by a user with a database of valid  
addresses, comprising: receiving the address from a source;  
accessing a database that contains one or more valid addresses;  
10 comparing the address from the user with the database of valid  
addresses; and storing information relating to the address in a  
companion file if a match is found..

15 In yet another embodiment, the invention is directed to a  
method for validating an address, comprising: receiving the  
address from a source, accessing a database of valid addresses,  
comparing the address from the source with the database of valid  
addresses, and storing selected information relating to the  
address in a companion file at the remote terminal if a match is  
found.

20 In still another embodiment, the invention is directed to a  
method of validating an address, comprising: receiving address  
data in a particular format; manipulating the data into a  
predetermined format corresponding to the particular format;  
comparing data in the predetermined format with valid addresses  
25 in the database; and presenting the results if one or more  
matches are found.

30 In another embodiment, the invention is directed to a method  
for importing one or more addresses from a database of addresses,  
where the database of addresses stores address data in a selected  
format. The method involves storing the one or more addresses  
at the database of addresses in the selected format; receiving  
the address data along with identification data to identify at  
least a characteristic of the database of addresses; processing  
the identification data; and processing the address data in a  
35 particular manner based on the identification data corresponding  
to the database of addresses.

In yet another embodiment, the invention is directed to a  
system for comparing a received address with a set of validated  
addresses. The system includes a remote device that maintains

1 an internal address book in a particular data format. The remote  
device includes at least one address book provider that is  
operative to interact with an external address book, the provider  
being operative to access data in the external address book and  
5 to provide data relating to the database structure format of the  
external address book. The system also includes an address  
matching server that maintains a database containing the set of  
validated addresses. The address matching server is capable of  
communication with the remote device via a communication network,  
10 and is operative in response to receipt of the address data and  
the database structure format data from the remote device to  
process the address data, compare the processed data with the set  
of validated addresses, and transmit one or more matches to the  
remote device.

15 In still another embodiment, the invention is directed to a  
method of comparing a received address with a set of validated  
addresses, including identifying an address in an external  
address book; accessing corresponding address data in the  
external address book with an address book provider that is  
operative to interface with the external address book; checking  
20 the address data against a local companion file to determine  
whether the address is valid; and validating the address at an  
address matching server that maintains a database of valid  
addresses if the result of the check with the local companion  
file does not validate the address.

25 In another embodiment, the invention is directed to a method  
of comparing a received address with a set of validated  
addresses, including identifying an address in an external  
address book; accessing corresponding address data in the  
external address book with an address book provider; and  
30 validating the address at a remote address matching server that  
maintains a database of valid addresses.

In yet another embodiment, the invention is directed to a  
method of comparing a received address with a set of validated  
addresses, including identifying an address in an external  
35 address book; accessing corresponding address data in the  
external address book with an address book provider; and  
validating the address at a server that maintains a database of

1 valid addresses, wherein validating comprises using one or more  
query permuters to process the address data.

#### BRIEF DESCRIPTION OF THE DRAWINGS

5 The objects, advantages and features of this invention will  
become more apparent from a consideration of the following  
detailed description and the drawings, in which:

10 FIG. 1 is a simplified block diagram of a remote user  
terminal ("client") connected to an address matching server via  
a two-way communication network according to one embodiment of  
the present invention;

FIG. 2 is an exemplary flow diagram of an address matching  
routine according to one illustrative embodiment of the  
invention;

15 FIG. 3 is an exemplary flow diagram of an address comparison  
routine according to one illustrative embodiment of the  
invention;

FIG. 4 is an address importing routine according to one  
illustrative embodiment of the invention;

20 FIG. 5 is a block diagram of the system of FIG. 1 and  
including plural external address book providers according to one  
illustrative embodiment of the invention;

FIG. 6 is a chart of various field names used by various  
external address books;

25 FIG. 7 is a schematic depiction of a direct permutation  
process to transform an input address format into a USPS-  
supported schema;

FIG. 8 is a schematic depiction of a single-line permutation  
process to transform an input address format into a USPS-  
supported schema;

30 FIG. 9 is a flow chart depicting the operational flow of a  
truncate permutation process to transform a portion of a direct  
permutation output into a new format;

FIG. 10 is a screen shot of a suitable user interface to  
allow a user to enter address information in a free format;

35 FIG. 11 is a screen shot of a typical user interface listing  
plural possible matches for an address validation request; and

FIG. 12 is a screen shot of a suitable user interface to  
allow a user to enter address information in a structured format.

## DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

1 Referring now to FIG. 1, there is shown an address matching system 10 according to one illustrative embodiment of the invention. The address matching system 10 includes an address matching system (AMS) server 12 that maintains a database 14 of  
5 a set of validated addresses. The AMS server 12 communicates with one or more remote terminals 16 (hereinafter referred to as "clients"). While only one client 16 is shown in FIG. 1, it will be understood that the system 10 is capable of interaction with multiple clients through a suitable interface. The clients 16  
10 and AMS server 12 engage in two-way communication over a suitable communication network 18. In one embodiment, communication network 18 comprises the Internet. It will be understood by those skilled in the art that the communication network may take many different forms, such as a local area network (LAN), wide  
15 area network (WAN), wired telephone network, wireless network, or any other network that supports data communication between respective entities.

The clients 16 may take many different forms, and in one illustrative embodiment comprise a personal computer and printer,  
20 with the personal computer in one embodiment being in communication with a PSD, either locally or over a communications network. Alternatively, the clients 16 may comprise computers or any other device that has processing capabilities and that may engage in communication over communication network 18. Other  
25 examples are wireless display devices, cellular telephones, and other mobile devices. The clients 16 are preferably programmed to provide functionality to allow users ("senders") to enter addresses in various format structures, and to compare those addresses with the database of valid addresses. The  
30 functionality of the clients 16 in verifying addresses is described in greater detail below in connection with FIGS. 2 - 4 and 9. The functionality of the client 16 and PSD in printing postage indicia in one illustrative embodiment is described in more detail in pending application Serial Number \_\_\_\_\_,  
35 entitled "VIRTUAL PRINTING OF INDICIA, LOGOS AND GRAPHICS", filed on August 29, 2000, the disclosure of which is hereby expressly incorporated by reference. An exemplary on-line system is described in U.S. patent Application No. 09/163,993 filed



1 September 15, 1998, the entire contents of which are hereby  
expressly incorporated by reference.

5 AMS server 12 will typically be associated with a web site  
that is accessible by each client 16 using a suitable Uniform  
Resource Locator (URL) address or the like, and will maintain a  
database 14 of valid addresses, which is accessed by the server  
12 in response to receipt of an address match request from a  
client 16, along with address data corresponding to the address  
that a sender at client 16 wishes to verify.

10 Referring now to FIG. 2, the basic operation of the address  
matching system 10 is described in more detail. Operation begins  
at step 30, with a sender entering a destination address for a  
corresponding piece of mail into the associated client 16. FIG.  
10 shows a suitable user interface 21 that client 16 may present  
15 to user for input of the pertinent shipping information. User  
interface 21 includes a pair of windows 22 into which the user  
may enter delivery and return addresses in a free form (i.e.,  
without separating the address information into multiple fields).  
The user interface 21 also includes a number of windows 23 into  
20 which the user may enter postage information, as is well known  
in the art.

At query block 32, client 16 determines whether the address  
has been previously validated. In one embodiment, this is  
accomplished by accessing a companion file maintained by the  
client 16 that stores selected information relating to each  
25 address that has previously been validated, and by comparing the  
address data with the data stored in the companion file. This  
process is described in greater detail below in connection with  
FIG. 3.

30 If the address has been previously validated, then operation  
proceeds to step 34, and the address is suitable for use and  
client 16 may use the address in a postage meter procedure. On  
the other hand, if the address has not been previously validated,  
then operation instead proceeds to step 36, and client 16  
35 processes the address and then makes an address matching request  
to AMS server 12, along with the processed address data.  
Processing of the address data can take many different forms, as  
is described in greater detail below. An example of such  
processing is described below in connection with FIG. 9.

1        Operation then proceeds to step 36, and AMS server 12  
compares the received address data with the data in database 14.  
As will be understood in the art, each record in database 14 will  
preferably consist of a plurality of elements (or fields), as  
will the received address data. Thus, AMS server 12 compares  
5        corresponding fields to determine whether one or more matches  
exists, and the results are transmitted to the requesting client  
16 at step 38. The results are preferably presented to the user  
as a list, such as the list shown in FIG. 11.

10        As used herein, the term "match" does not require that the  
address data correspond identically to the data in database 14.  
Rather, a match (or at least a potential match) may be declared  
where there is a substantial similarity between the address data  
and the data in one or more records. As is well known in the  
art, a match can be declared when the two sets of data exceed  
15        some preset threshold. For example, a match can be declared  
where each field exceeds some preset threshold. Alternatively,  
a match can be declared where certain of the fields match  
identically (such as the field that contains the city data) and  
where the other fields exceed some threshold matching value.  
20        Thus, it will be understood that a match can be declared in many  
different ways.

      Operation then proceeds to query block 40, and client 16  
determines whether there are more than one match in the results.  
If not, then operation proceeds to step 42, and the matched  
25        address may then be used by the client 16 in a postage meter  
procedure. If, on the other hand, the AMS server 12 returned  
more than one potential match, then operation proceeds to step  
44, and the sender selects one of the potential addresses.  
Client 16 then prepares an address match request with the address  
30        data from the selected address. The query is transmitted to AMS  
server 12. At step 46, AMS server 12 processes the request, and  
transmits all necessary information relating to the single  
address to client 16, which may then use the address in a postage  
meter procedure. Operation then terminates.

35        In one embodiment, at step 46 the AMS server 12 employs an  
address filter to ensure that only a single address is returned  
to client 16 (i.e., in a match address single query). The filter  
only allows an address that exactly matches the input address to

1 proceed to client 16. If there is no exact match, then the  
filter blocks all addresses except for a default address, as  
determined by server 12. When the filter analyzes a returned  
range-based address, it initially determines whether the contents  
5 of that result, aside from the range, match the corresponding  
contents in the original input address. If so, the filter  
determines whether the ranged value lies within the range in the  
result address. If so, then the filter considers the address  
identical and server 12 passes the address back to client 16.

10 On the other hand, if the contents of the result, aside from  
the range, do not match the corresponding contents in the  
original input address, then the filter only passes a default  
address.

15 When the server 12 returns the results to client 16, some of  
the addresses may contain ranges (e.g., 100 Main Street Suite  
100-105). If that address is the one selected by the user,  
client 16 then requires user to select the correct number from  
the range (e.g., Suite 102). That address is then transmitted  
back to server 12 for cleansing and validation. For this query,  
20 client 16 utilizes at "Matching Address Single" query for  
validation, and the results ensures a return of a Delivery Point  
Bar Code (DPBC) value. The "Matching Address Single" query is  
described in greater detail below.

25 In one embodiment, AMS server 12 returns the results of an  
address validation along with a result code to indicate to client  
16 the outcome of the validation process. For example, the  
result code may indicate that the input address 1) is valid, 2)  
is corrected, 3) was not found, 4) has multiple potential  
matches, 5) includes an invalid number or street, 6) includes an  
invalid city, 7) includes an invalid state, 8) includes an  
invalid ZIP code, and the like. The result code may then be used  
30 by client 16 accordingly.

35 In this manner, system 10 provides a server-based address  
matching system that does not require each user to maintain a  
database of validated addresses. As the set of validated  
addresses changes, only the central database 14 at server 12  
requires updating. Each individual client 16 need not update any  
list in their respective memories. In this manner, system 10  
ensures that addresses may be properly validated as soon as

1 database 14 is modified. There is no need to rely upon each individual user at each client 16 to update the respective machines.

5 As described above, each client 16 preferably maintains one or more companion files (e.g., data files, a database, or any other suitable storage location in a memory) to which is saved selected data for each validated address. In one embodiment, the records in the companion file are purged when the client 16 software determines that the record expiration date has expired, thereby requiring that the previously validated addresses be  
10 validated again. In another embodiment, each record includes an associated expiration date, so that client 16 may determine whether the record is valid or whether it has expired, such that the address must be validated through AMS server 12.

15 In one illustrative embodiment, client 16 maintains a companion file for each address book with which client 16 interacts. Thus, client 16 maintains a companion file for its own internal address book, and plural companion files for respective external address books, as is described in more detail below. Each address book generates a unique address identity for  
20 each address record. The address identity data is stored in the companion file as a data record. In one embodiment, each record in the companion file also includes a plurality of data members, including: 1) ZIP code, 2) Address Hash, 3) Address Hash Size, 4) Address Identity, 5) Expiration date, and 6) Archive Version. The ZIP code is preferably the full twelve-digit ZIP code (ZIP  
25 plus add-on plus delivery point plus check digit), or the five-digit ZIP code if the AMS server 12 did not assign an add-on to the validated address.

30 The Address Hash is the hash value for the last two printed lines (all in upper case) of the address when it was validated. As is well known in the art, a hash value is a number generated from a string of text. The hash value is substantially smaller than the text itself, and is generated by a formula in such a way that it is extremely unlikely that some other text will produce  
35 the same hash value. The Address Hash size is the actual size, in bytes, of the Address Hash value.

The Address Identity data member is a key that uniquely identifies the address to which the companion record corresponds

1 in an address book database, to associate the address in an address book with the companion record in the companion file for that address book. As described above, client 16 preferably maintains a companion file for each address book.

5 The Expiration date, in one embodiment, corresponds to the expiration date of the AMS database used to cleanse the address. The AMS server 12 provides this date as part of the output of each address matching transaction. The expiration date is the date at which the address validation expires for the  
10 corresponding record. After that date, the address is no longer considered valid, even if the user has not altered it and the hash values match.

The Archive Version is a version number for the respective record format. If a change is made to the companion record  
15 format, the software can identify "old" records and "new" records based on the Archive Version information, which is used to ease future changes to the format of companion file records. In one embodiment, the Archive Version number is used in conjunction with the Microsoft® Foundation Classes object serialization  
20 feature (which is used to read and write companion files).

Referring now to FIG. 3, operation of system 10, and in particular client 16, in connection with validating an address from an address book is described in more detail. Operation  
25 begins at step 60, with the user selecting a desired destination address from an address book at client 16. At query block 62, client 16 determines whether there is a corresponding record in the companion file for the selected address. In one embodiment, this is accomplished by accessing the Address Identity values in the companion file to determine whether there is an associated  
30 record for the selected address. If not, operation proceeds to step 64, and the selected address must be validated through AMS server 12. System 10 then proceeds through the process shown in FIG. 2 and described above.

On the other hand, if there is an associated record in the  
35 companion file, the operation proceeds to query block 66, and client 16 determines whether the record in the companion file is expired, for example, by reviewing the Expiration date data member stored in the companion file record. If the record has expired, then the record is stale and operation proceeds to step

1 64. If the record is instead extant, then operation proceeds to  
query block 68, and client 16 determines whether the address has  
been altered by the user. Preferably, this is accomplished by  
comparing hash values for the selected address and the record in  
5 the companion file. In one embodiment, the hash value is  
generated from the last two lines of the address (e.g., street  
and city/state/ZIP), and the hash value is compared to the  
recorded hash value in the record. If the hash values match,  
then operation proceeds to step 70, and the selected address is  
10 still valid. If the hash values do not match, then operation  
proceeds to step 64, and the address must be validated through  
AMS server 12.

In one embodiment, each client 16 maintains an internal  
address book (i.e., a data file) that contains address records  
15 stored in a preselected format. For example, the address book  
may include the following element fields for each record, in a  
specific order: Name, Title, Street Address, City, State, and Zip  
Code. This order is the internal canonical schema supported by  
system 10.

20 System 10 also provides software resident at each client 16  
that allows for the importation of address data from various  
third-party address books into the address book maintained by  
client 16. In one illustrative embodiment, client 16 maintains  
a plurality of address book providers 15 (FIG. 5), with each  
25 provider 15 being operative to interface with a corresponding  
external address book application. In another embodiment, for  
example, in the case of address books that do not have an  
associated provider 15, the importation of addresses relies on  
comma-separated value (CSV) files. As is understood by those  
30 skilled in the art, many software applications can generate CSV  
files that contain data in fields separated by commas.

35 Operation of the system 10 in importing CSV files is now  
described in connection with FIG. 4. Operation commences at step  
80, with a user at client 16 requesting to import one or more  
addresses from a third-party address book. At step 82, the user  
selects the CSV file(s) to import from a list of CSV files stored  
in memory, or in any other suitable manner. Client 16 then  
prompts the user to select the address book application used to  
store the CSV file, and at step 84 the user enters such

1 information. Client 16 maintains a set of data corresponding to  
field types and field sequence for each address book application.  
For example, one address book application may designate five  
fields for each address record in a selected order, such as name,  
5 organization, street address, city and state, and ZIP code.  
Another address book application may include the name and  
organization in one field, and separate the city and state into  
different fields. Thus, client 16 is preferably programmed with  
such information for various address book applications.

10 Operation then proceeds to step 86, and client 16 receives  
the CSV file(s) from the file system, and also retrieves the  
appropriate format information based on the address book  
application identified by the user. Client 16 then parses  
through the data in the CSV file(s), reading the data for the  
15 respective fields and inserting the data into a format that is  
supported by client 16 and that is suitable for storage in the  
native address book database of system 10. As is known in the  
art, the USPS address matching service supports a particular  
schema format (FIG. 7), in which the first field contains data  
20 relating to Address Line 1 (e.g., street address), the second  
field contains data relating to Address Line 2 (e.g., Post Office  
Box number) and a firm or organization name, the third field  
contains data relating to the city, the fourth field contains  
data relating to the state, and the fifth field contains data  
25 relating to the ZIP code.

Thus, client 16 determines, based on the address book  
application used to create the address record, which portion of  
the CSV data maps to the native address book database supported  
by system 10. Then, this data is mapped to canonical fields as  
30 all other address book databases are accessed through the  
respective address book providers, as is described in more detail  
below.

Operation then proceeds to query block 88, and client 16  
determines whether the user wishes to validate one or more of the  
35 addresses via AMS server 12. If so, then system 10 performs the  
validation process as is described above in connection with FIG.  
2. If not, then operation terminates.

If the user wishes to validate an address, client 16  
determines whether a match has been found for each address that

1 the user wishes to validate. As described above, the validation  
process may include transmitting a list of potential matches to  
client 16 for display to the user, and then allowing the user to  
select from that list to create a match. If a match is found,  
5 the address data is added to the address book maintained by  
client 16, and information relating to the matched address is  
stored in the companion file, as described above.

As described in more detail below, when client 16 is  
provided with a list of potential address matches, the user then  
may select one of the addresses from the list. Once the user  
10 selects one of the addresses, server 12 is queried again using  
one or more permuters, as are described in more detail below.  
Server 12 then returns a validated, cleansed address that is  
stored in the appropriate address book, and the appropriate  
updates are made to the corresponding companion file.

15 On the other hand, if there are no matches found for one or  
more of the addresses, client 16 determines whether the user  
wishes to correct the address information for the unapproved  
address or addresses, by displaying a suitable graphic on the  
screen and allowing the user to decide whether to correct the  
20 address(es) or not. If not, then the non-approved address data  
is discarded. If the user does wish to correct an address, then  
client 16 receives such information from the user. In one  
embodiment, client 16 displays a suitable graphic with plural  
fields (windows) into which the user may enter or modify data,  
25 as is shown in FIG. 12. Once the user has entered the changes,  
client 16 once again attempts to validate the address via AMS  
server 12.

Referring now to FIG. 5, there is shown another illustrative  
embodiment of system 10. In this embodiment, system 10 maintains  
30 one or more external address book application programming  
interface (API) providers 15 (hereinafter "address book  
providers"). Each address book provider 15 is designed to  
natively support a corresponding external address book, such that  
the address data at each external address book need not be stored  
35 in CSV (or some other) format prior to introduction to client 16.  
In one illustrative embodiment, the address book providers 15 are  
COM components that are installed by client 16. For example, one  
or more of the address book providers 15 are built on Extended



1 Systems' Harmony translator API that enables direct address book  
database traversal. Other providers 15 may be based on Messaging  
Application Programming Interface (MAPI) protocol. When each  
address book provider 15 is installed, the provider registers in  
5 a suitable registry, such as Windows registry, using the COM  
category API.

Each address book provider 15 is designed to interface  
between the client address book component 15 and the respective  
external address book component 17, which allows for dynamically  
10 modifying data in the corresponding address book, without the  
need for making a copy of the data contained in the external  
address book. Each provider 15 allows access to the database or  
databases of the respective external address book, and provides  
the mechanism for translating, in both directions, between the  
15 external address book's database structure and the internal  
canonical schema of system 10 (e.g., data or other instructions  
that serve to identify the pertinent characteristics of the  
external address book). This then allows for entering addresses  
through system 10, and writing them to an external address book  
20 17, or reading address data from one or more external address  
books 17 and processing the address data at server 12 to map the  
address data into a data structure that has the appropriate  
format for comparison with the set of validated addresses  
maintained by server 12.

25 When an address book provider 15 is loaded, it reports its  
schema representation to client 16, and for each field in its  
schema, the provider communicates 1) the name of the field, 2)  
a canonical field name to which the field is mapped, 3) data type  
(e.g., string), 4) data size, and 5) attributes of that field.  
30 Thus, when address data is read from a particular address book  
and such data is made available to client 16, client 16 may pass  
that information on to server 12, along with schema  
representation information to server 12, so that server 12 may  
intelligently process the incoming address data.

35 The internal address book component 19 of client 16 provides  
a canonical schema into which the addresses from the respective  
external address books are formatted. The canonical schema is  
the symbolic representation of all postal address fields that are  
encountered in the respective external address books. These

1 fields identify the semantics of the postal fields used for a particular address book. FIG. 6 depicts the canonical field names encountered for various address books.

5 As described above, client 16 transmits an address validation request to AMS server 12, along with the address data in a format that is compatible with the internal canonical schema. However, it will be understood by those skilled in the art that a match may still be hard to obtain in the first stage, because of different formats that AMS server 12 must attempt to map to the USPS-supported format.

10 In order to increase the likelihood of obtaining one or more matches, system 10 provides a set of query permutations to configure the input address in various formats to enhance the likelihood of obtaining one or more matches. In one illustrative embodiment, system 10 utilizes three different permuting techniques to map different schema formats to the USPS-supported schema, namely a direct permuter, a single line permuter, and a truncate permuter. Thus, a structured address is processed, prior to transmission to AMS server 12, using the three permuters to increase the chance of finding a match for the address. Each permuter attempts to map the available input fields to the USPS-supported schema, but with a slightly different form. It will be understood that more or less permuters can be used.

25 The direct permuter directly maps the input data to the USPS-supported schema. FIG. 7 schematically shows a direct permutation process performed on an address record consisting of three address line fields, in which the order of the first and second fields is switched, and the third field (Address Line 3) is mapped to respective City, State, and ZIP code fields in the internal canonical schema of system 10.

30 The single line permuter is used when the street address information is not contained on a single line, for example, when the suite number is on its own line. The single line permuter appends the Address Line 2 data to Address Line 1 to create a single line for street address data. FIG. 8 schematically shows such a permutation process.

35 The truncate permuter receives the direct permuter output and analyzes the first address line to determine whether to modify the data in that line. FIG. 9 is a flow chart showing the

1 process steps of the truncate permuter. Operation begins at step  
100, and the first address line from the direct permuter output  
is taken. At query block 102, client 16 determines whether there  
are three or fewer tokens in the address line. As used herein,  
5 a "token" is defined as an element that is separated from other  
elements by commas, periods, tabs, spaces, apostrophes, quotes,  
colons, or semicolons.

If there are three or fewer tokens in the line, then  
operation proceeds to step 104, and the truncate permuter does  
10 not modify the address line, and operation ends. On the other  
hand, if there are more than three tokens in the address line,  
then operation instead proceeds to step 106, and client 16  
compares the fourth through last tokens in the address line with  
a list of common abbreviations. For example, the address line  
15 may be "123 South Main St.". The fourth token is "St", which is  
found in the list of common abbreviations. The list of common  
abbreviations will also include other well-known abbreviations,  
such as "Ave", "Blvd", "NW", "SW", and the like.

Then, at query block 108, client 16 determines whether there  
20 is a match. If not, operation proceeds to step 110, and only the  
first three tokens are kept, and the fourth through last tokens  
are deleted from the address line. If there is a match in one  
of the fourth through last tokens, then operation instead  
proceeds to step 112, and all tokens after the matched token are  
25 dropped. For example, "123 South Main St. Harbor Marina" becomes  
"123 South Main St.". This modified address is then added to the  
ordered, permuted address list for validation at AMS server 12.

As described above, each external address book has its own  
unique set of fields, some of which are not mapped to the USPS-  
30 supported schema. The data in those fields is preferably saved  
by AMS server 12 in memory during the permutation and address  
validation process. Then, when a validated address is returned  
to client 16, the data is reincorporated into the final address.  
For example, the external address book may have fields for Title  
35 and Department, which are not contained in the USPS-supported  
schema. Thus, the data in those fields is removed during the  
conversion into the USPS-supported schema. Once a valid address  
is returned, the data is reincorporated into the address that is  
presented to the user.

1 Typically, when multiple matches are returned to client 16,  
the results do not include Delivery Point Bar Code (DPBC) values.  
Thus, when a user selects one of the addresses, a second query  
is transmitted to AMS server 12 to retrieve all necessary  
5 information concerning the selected address, including a DPBC  
value. In another embodiment, AMS server 12 is programmed to  
return a DPBC value for each output address transmitted to client  
16. In yet another embodiment, AMS server 12 compares the input  
9-digit ZIP code with the ZIP code of each output address. For  
10 each output address whose ZIP code matches the input ZIP code,  
the DPBC value is retrieved and transmitted for each output  
address.

In one embodiment, system 10 ranks each of the potential  
matches according to the following criteria: 1) when only a  
15 single address is returned, no ranking is necessary; 2) all  
multiple addresses are returned with a default ranking of  
"possible"; 3) addresses that match the firm name or ZIP code are  
ranked as "better". Ranking is preferably performed by a series  
of conventional ranking agents, such as those that determine if  
20 an address matches the input address Firm Name or ZIP code.

From the foregoing, it will be apparent to those skilled in  
the art that the system and method of the present invention  
provide for validating addresses through a central server. The  
system and method also provide novel techniques for improving the  
25 outcome of such address validating procedures.

While the above description contains many specific features  
of the invention, these should not be construed as limitations  
on the scope of the invention, but rather as exemplary  
embodiments thereof. Many other variations are possible.  
Accordingly, the scope of the invention should be determined not  
30 by the embodiments illustrated, but by the appended claims and  
their legal equivalents.

## WHAT IS CLAIMED IS:

1. A system for comparing a received address with a set of validated addresses, the system comprising:

a remote device that is operative to receive address data from a source; and

an address matching server that maintains a database containing the set of validated addresses, wherein the address matching server is capable of communication with the remote device via a communication network, the address matching server being operative in response to receipt of the address data from the remote device to process the address data using at least one query permuter, compare the processed data with the set of validated addresses, and transmit one or more matches to the remote device.

2. The system of claim 1, wherein the source is a user manually entering address data into the remote device.

3. The system of claim 1, wherein the source is a database of addresses that is accessible by the remote device.

4. The system of claim 1, wherein the remote device and address matching server are capable of communication over a computer network.

5. The system of claim 4, wherein the computer network comprises the Internet.

6. The system of claim 1, wherein the server is operative to process the received address data to generate multiple data structures incorporating the address data.

7. The system of claim 6, wherein the server applies a plurality of query permuters to the address to generate the multiple data structures.

8. The system of claim 1, wherein the remote device is operative to maintain a companion file that contains information

1 relating to validated addresses, and wherein the remote device  
is operative to write said information to the companion file.

5 9. The system of claim 1, wherein the remote device  
comprises a computer terminal.

10 10. The system of claim 1, wherein the remote device  
comprises one of a cellular telephone and wireless display  
device.

10 11. A method for validating an address entered by a user at  
a terminal, comprising:

receiving the address from a source;

accessing a database that contains one or more valid  
addresses;

15 comparing the address from the user with the database of  
valid addresses; and

storing information relating to the address in a companion  
file if a match is found.

20 12. The method of claim 11, wherein receiving the address  
comprises importing the address from a database of addresses.

25 13. The method of claim 12, wherein the address from the  
database of addresses is saved as comma-separated value (CSV)  
data, and further including:

determining selected characteristics of the database of  
addresses; and

30 processing the CSV address data based on the characteristics  
of the database of addresses.

14. The method of claim 11, further including:

receiving a second address from the source;

35 comparing selected information from the second address with  
stored information in the companion file;

approving the address for use if the selected information  
corresponds with the stored information in the companion file;  
and

1       accessing the database of valid addresses if no match is  
found.

5       15. The method of claim 14, wherein comparing the second  
address with the stored addresses in the companion file comprises  
determining whether a stored address in the companion file is  
stale, and rejecting the stored address if it is stale.

10       16. The method of claim 11, wherein accessing the database  
comprises accessing a remote database over a communication  
network.

15       17. The method of claim 16, wherein the database is  
maintained by a remote address matching server.

20       18. A method for matching an address with a database of  
valid addresses, comprising:

      receiving address data in a particular format;

      manipulating the data into a predetermined format  
corresponding to said particular format;

      comparing data in the predetermined format with valid  
addresses in the database; and

      presenting the results if one or more matches are found.

25       19. The method of claim 18, wherein comparing data in the  
predetermined format with valid addresses in the database  
comprises accessing a remote database of addresses over a  
communication network.

30       20. The method of claim 19 wherein the database is  
maintained by a remote address matching server.

35       21. The method of claim 18, wherein manipulating comprises  
applying a plurality of query permuters to the address data to  
convert the data into respective formats.

      22. The method of claim 21, wherein applying a plurality of  
query permuters comprises applying at least one of a direct  
permuter and a single line permuter to the address data.

1        23. The method of claim 22, wherein applying further  
comprises applying a truncate permuter to the output structure  
of the direct permuter.

5        24. A method for importing one or more addresses from a  
database of addresses, where the database of addresses stores  
address data in a selected format, the method comprising:

storing the one or more addresses at the database of  
addresses in the selected format;

10        receiving the address data along with identification data to  
identify at least a characteristic of the database of addresses;

processing the identification data; and

15        processing the address data in a particular manner based on  
the identification data corresponding to the database of  
addresses.

25        25. The method of claim 24, wherein storing the one or more  
addresses in a particular format comprises storing the one or  
more addresses in a comma-separated value (CSV) format.

20        26. The method of claim 24, wherein receiving the address  
data along with identification data comprises receiving the  
address and identification data from an external database of  
addresses.

25        27. The method of claim 24, wherein receiving the  
identification data comprises receiving identification  
information input by a user.

30        28. A system for comparing a received address with a set of  
validated addresses, the system comprising:

35        a remote device that maintains an internal address book in  
a particular data format, the remote device including at least  
one address book provider that is operative to interact with an  
external address book, the provider being operative to access  
data in the external address book and to provide data relating  
to translation between a database structure of the external  
address book and a second format; and



1 an address matching server that maintains a database  
containing the set of validated addresses, wherein the address  
matching server is capable of communication with the remote  
device via a communication network, the address matching server  
5 being operative in response to receipt of the address data and  
translation data from the remote device to process the address  
data and transmit one or more matches to the remote device.

10 29. The system of claim 28, wherein the remote device  
includes at least two address book providers, wherein the address  
book providers are compatible with respective external address  
books.

15 30. The system of claim 28, wherein the address book  
provider is operative to provide schema representation  
information to the server.

20 31. The system of claim 30, wherein the address book  
provider is operative to provide information for plural fields  
in the schema representation.

25 32. The system of claim 31, wherein the address book  
provider provides, for each field in the schema, one or more of  
the name of the field, a canonical field name to which the field  
is mapped, data type, data size, and attributes of that field.

33. A method of comparing a received address with a set of  
validated addresses, the method comprising:

identifying an address in an external address book;

30 accessing corresponding address data in the external address  
book with an address book provider that is operative to interface  
with the external address book;

checking the address data against a local companion file to  
determine whether the address is valid; and

35 validating the address at an address matching server that  
maintains a database of valid addresses if the result of the  
check with the local companion file does not validate the  
address.

1       34. The method of claim 33, wherein validating the address  
at the address matching server comprises providing the address  
data and information relating to the external address book to the  
server, and wherein the server is operative to process the  
5       address data in a particular manner based on the contents of said  
information.

10       35. The method of claim 33, wherein checking the address  
data against a local companion file comprises determining whether  
a record in the companion file is extant.

      36. The method of claim 33, wherein checking the address  
data against a local companion file comprises comparing the  
address data with a hash value stored in the companion file.

15       37. The method of claim 33, wherein validating comprises  
validating the address at a remote address matching server.

20       38. A method of comparing a received address with a set of  
validated addresses, the method comprising:  
      identifying an address in an external address book;  
      accessing corresponding address data in the external address  
book with an address book provider; and  
      validating the address at a remote address matching server  
25       that maintains a database of valid addresses.

30       39. The method of claim 38, wherein validating the address  
at the remote address matching server comprises providing the  
address data and information relating to the external address  
book to the server, and wherein the server is operative to  
process the address data in a particular manner based on the  
contents of said information.

35       40. The method of claim 39, wherein the address matching  
server utilizes one or more query permuters to process the  
address data.

1       41. The method of claim 40, wherein applying a plurality of  
query permuters comprises applying at least one of a direct  
permuter and a single line permuter to the address data.

5       42. The method of claim 40, wherein the address matching  
server utilizes a plurality of query permuters to process the  
address data.

10       43. The method of claim 42, wherein applying a plurality of  
query permuters comprises applying at least one of a direct  
permuter and a single line permuter to the address data.

15       44. The method of claim 43, wherein applying further  
comprises applying a truncate permuter to the output of the  
direct permuter.

20       45. A method of comparing a received address with a set of  
validated addresses, the method comprising:  
      identifying an address in an external address book;  
      accessing corresponding address data in the external address  
book with an address book provider; and  
      validating the address at a server that maintains a database  
of valid addresses, wherein validating comprises using one or  
more query permuters to process the address data.

25       46. The method of claim 45, wherein validating comprising  
utilizing a plurality of query permuters.

30       47. The method of claim 46, wherein applying a plurality of  
query permuters comprises applying at least one of a direct  
permuter and a single line permuter to the address data.

35       48. The method of claim 45, wherein validating the address  
comprises validating the address at a remote address matching  
server.

FIG. 1

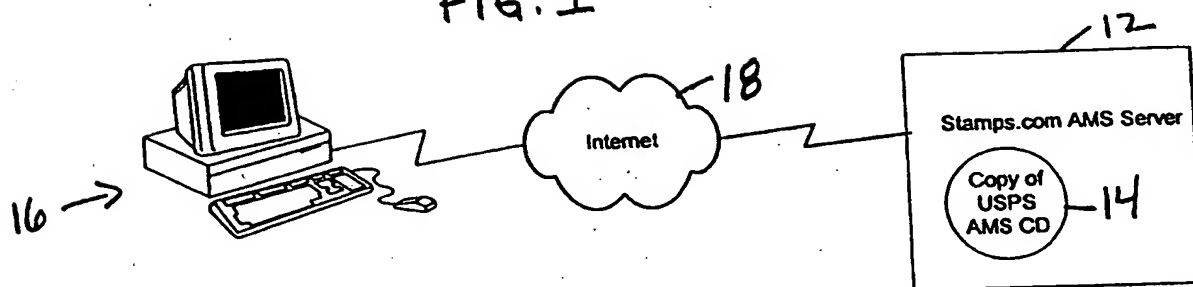


FIG. 7

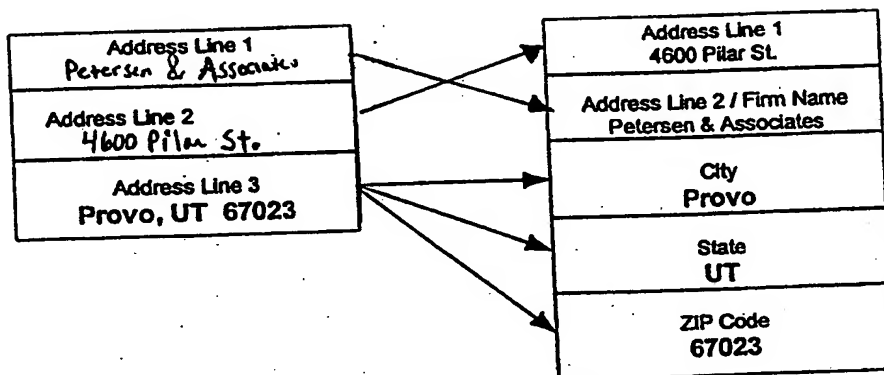
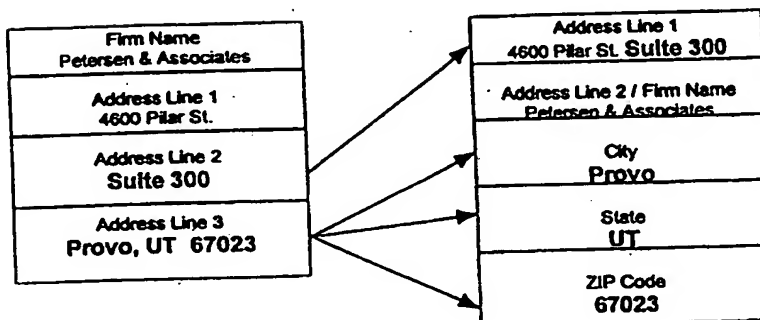
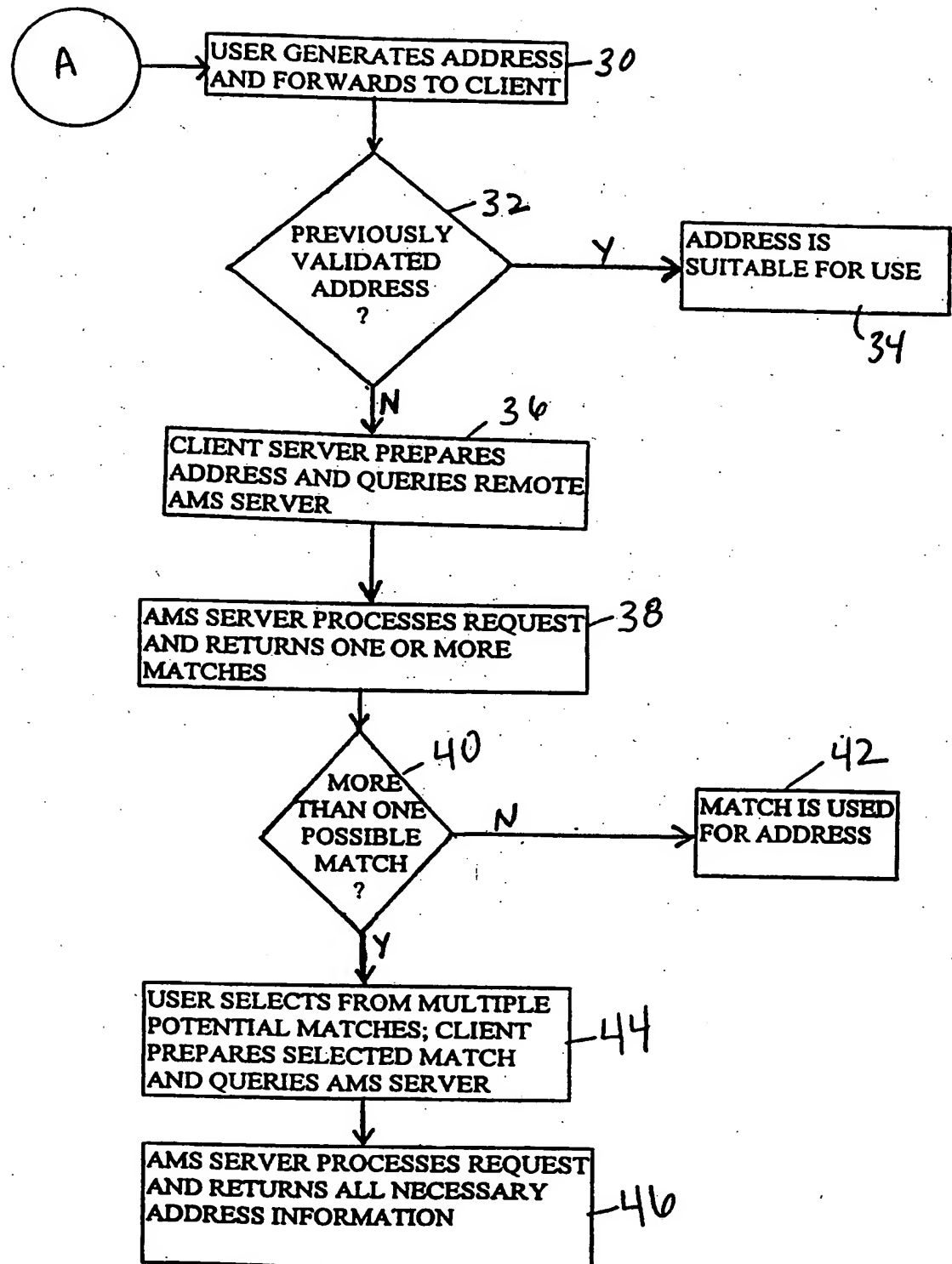


FIG. 8



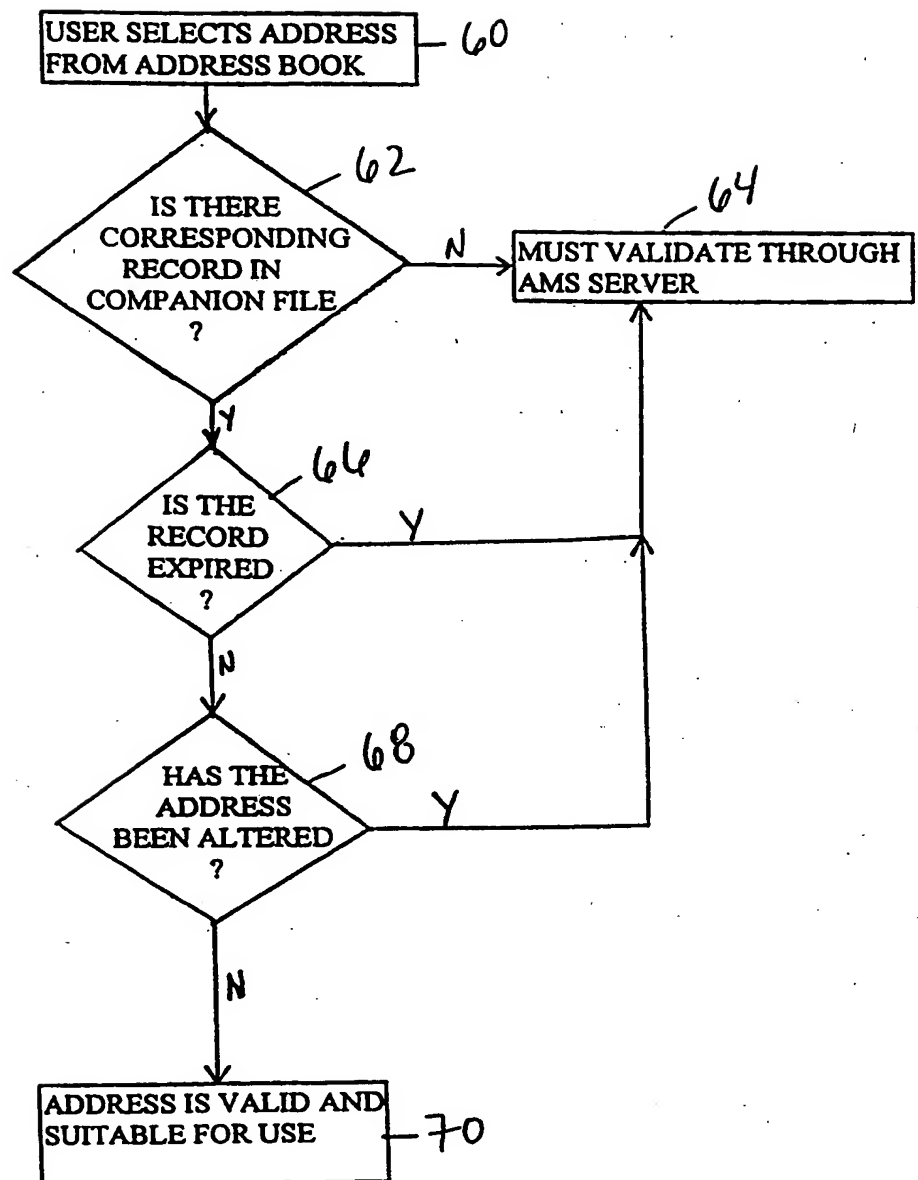
2/9

FIG. 2



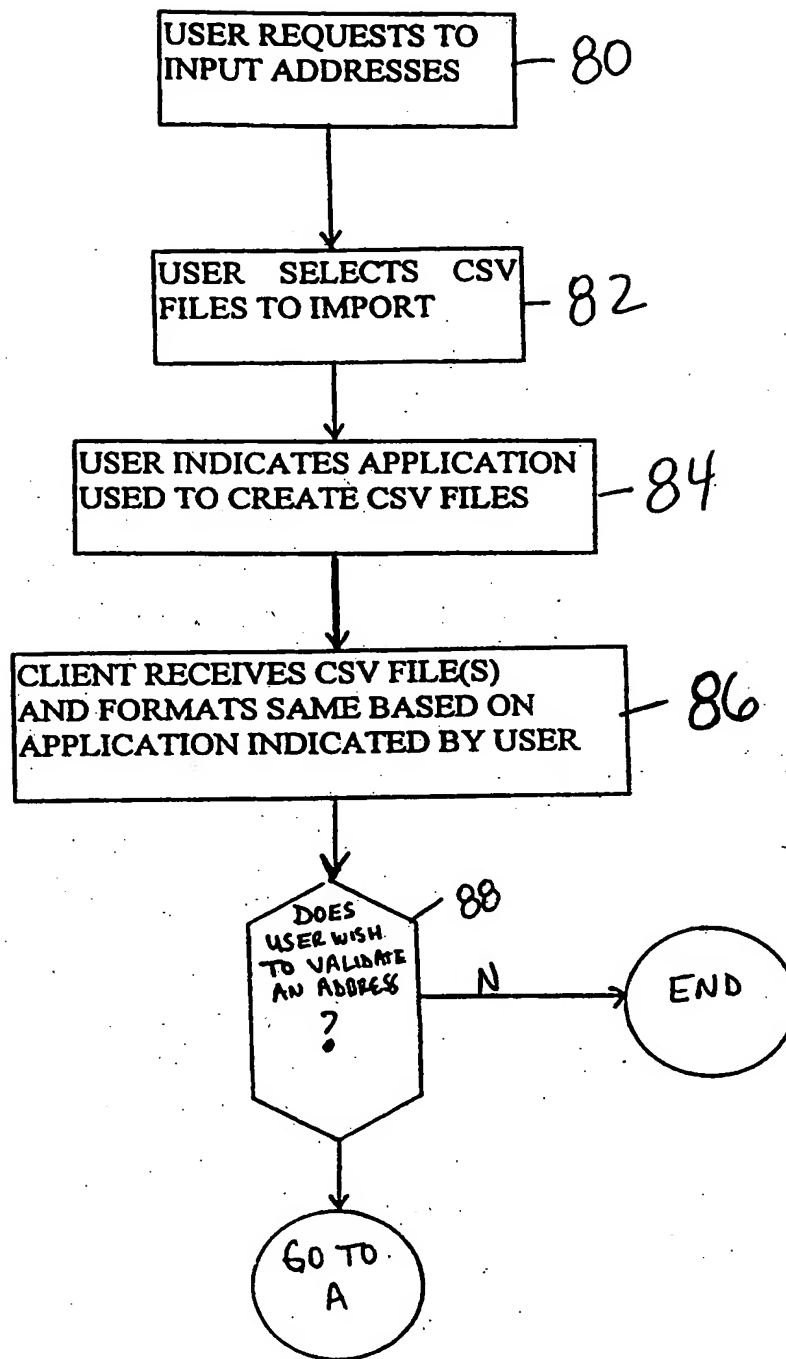
3/9

FIG. 3



4/9

FIG. 4



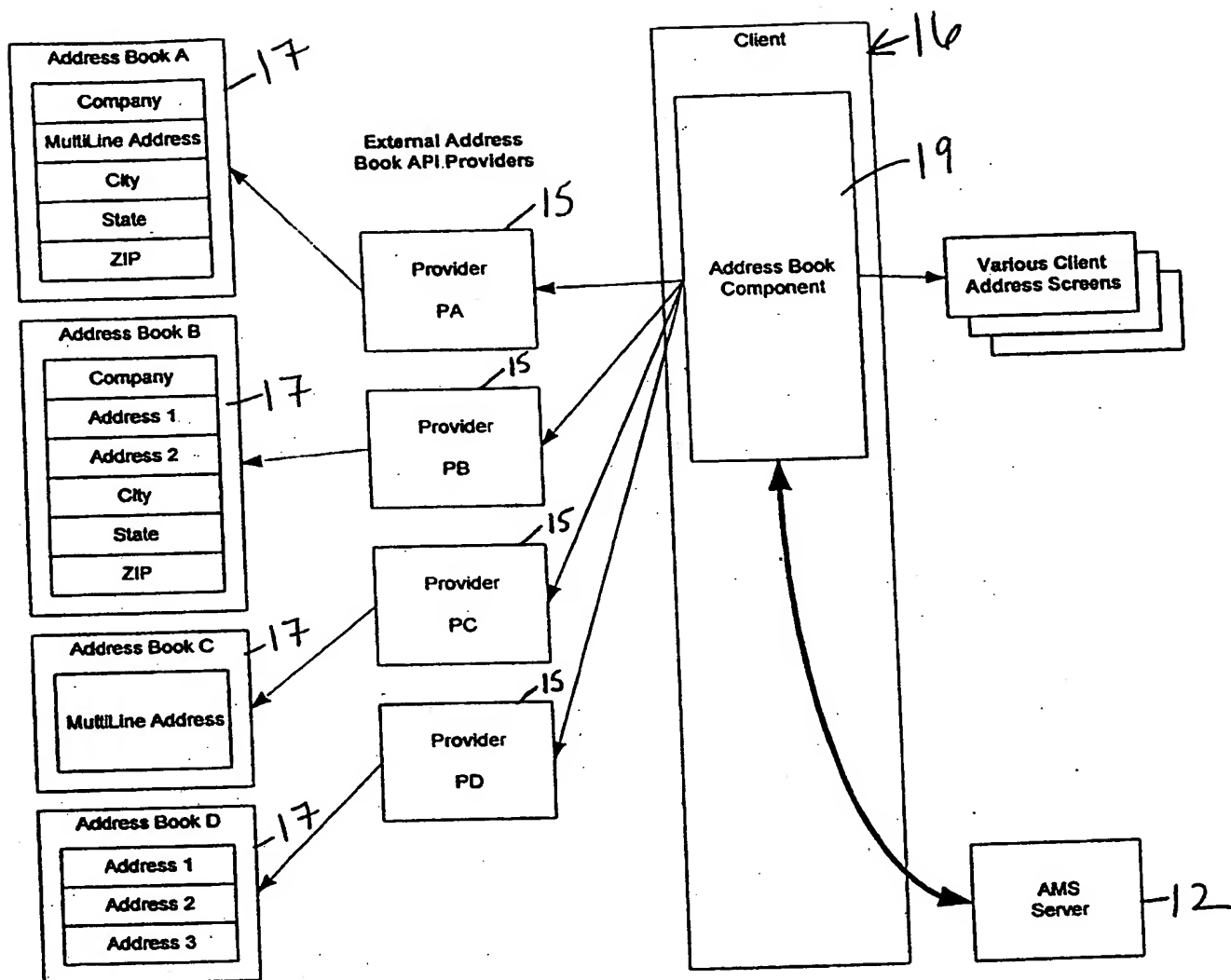


FIG. 5

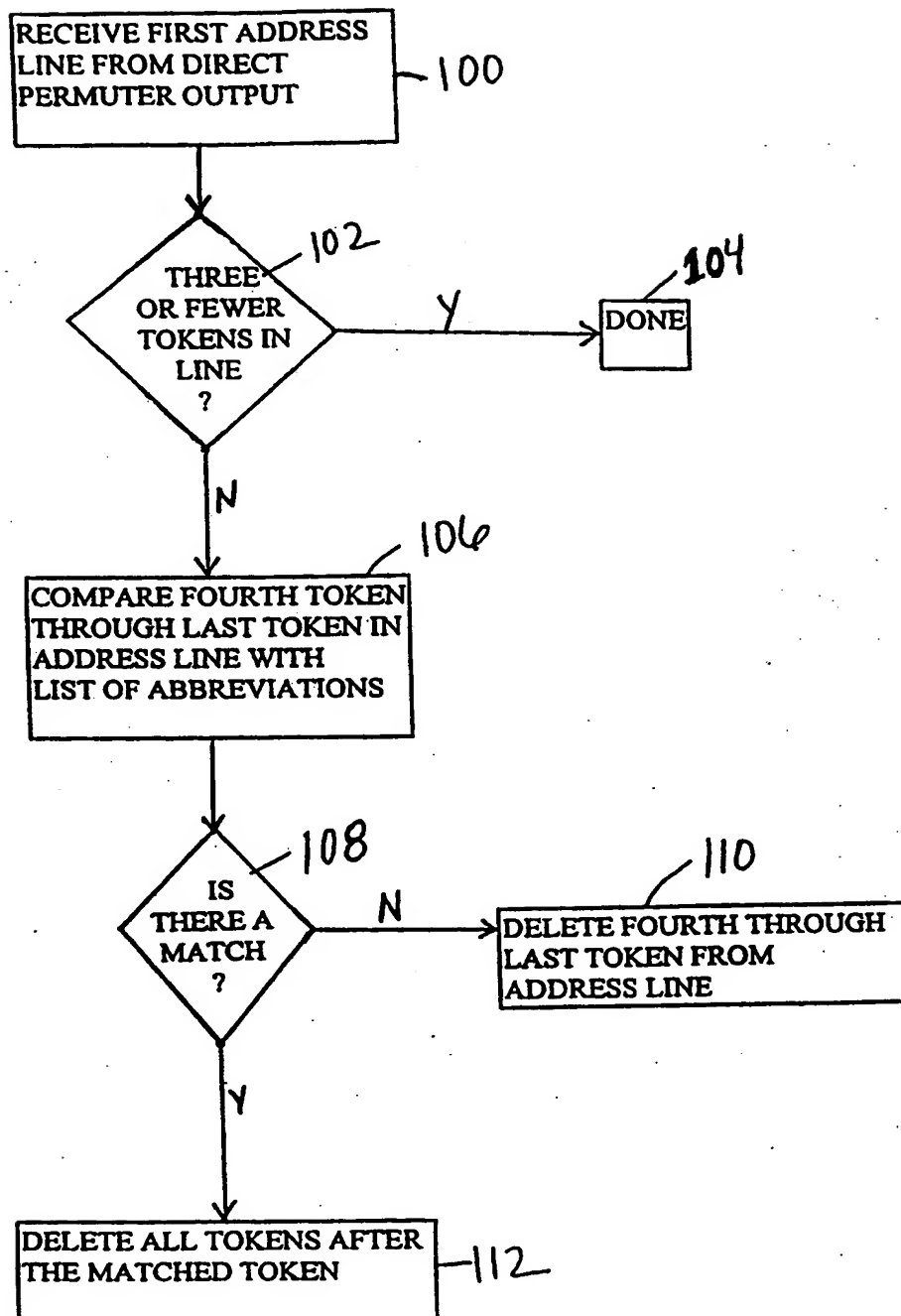


Canonical Field Name	Notes
None	
DisplayName	
FullName	Mutually exclusive with [NamePrefix, NameSuffix].
NamePrefix	Mr., Ms., etc.
FirstName	
MiddleName	
LastName	
NameSuffix	Jr., M.D., etc.
Title	Corresponds to USPS Pub 28's "FUNCTIONAL TITLE".
Department	
Company	
MultiLineAddress	Mutually exclusive with [Address1-3, Unit].
Address1	
Address2	
Address3	
UnitName	Apt, Suite, etc. Cannot exist if Unit doesn't exist.
Unit	Can exist only when Address1 is present but no Address2 or Address3.
City	
State	
Zip	5-digit or 9-digit ZIP code if there is no ZipAddOn field.
ZipAddOn	4-digit extension to the 5-digit ZIP code.
Country	
DPB	Delivery Point Barcode. Must be a 2 or 3 digit number.
CheckDigit	Must be a 1 digit number.
Mailstop	A USPS code.
Route	A USPS code.
Cargo	Non-user-visible BLOB that Stamps.com can "own".

FIG. 6

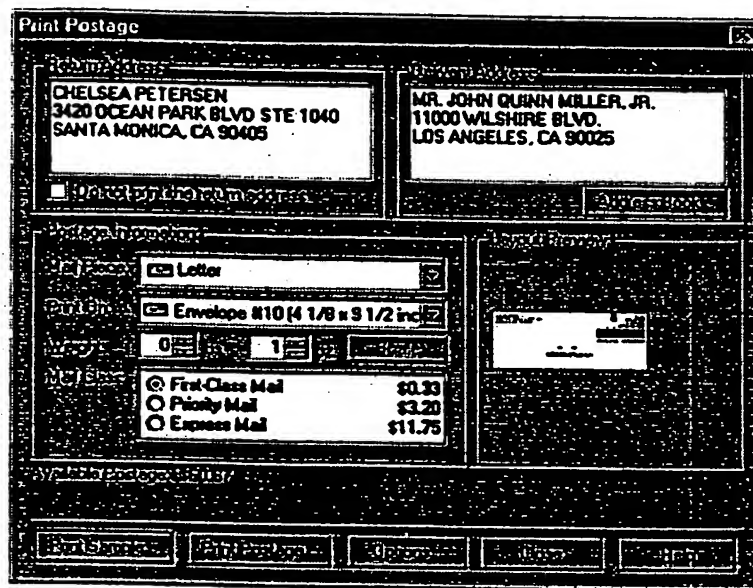
7/9

FIG. 9



PCT/U  
BEST AVAILABLE COPY

8/9



← 21

FIG. 10

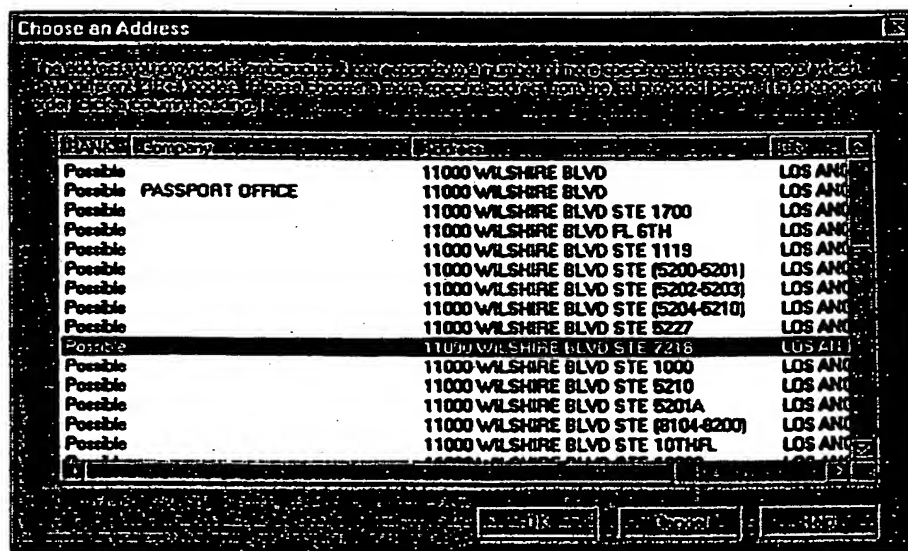


FIG. 11

**Contact Information**

For information or to contact us, after filling in  
the appropriate box, check one of the following boxes.  
If you are not interested in our products, please do not  
fill in any of the following boxes.

Name:	Mr. <input checked="" type="checkbox"/> Mrs. <input type="checkbox"/>	Amos	T	Lewis	
Title:	DIRECTOR OF MARKETING				
Company:	ATLAS CORP.				
Address:	14200 REGENTS WAY				
City:	ALAMEDA				
State:	CA	Zip Code:	90456-1212		

OK Cancel Print Mail

# INTERNATIONAL SEARCH REPORT

Inte. Jional Application No

PCT/US 00/41322

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 7 G07B17/00

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G07B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 663 652 A (PITNEY BOWES) 19 July 1995 (1995-07-19)	1-4, 6-9, 11, 12, 14-24, 26-31, 33-48
Y	claim 1; figures 1, 2	5, 10, 13, 25
Y	EP 0 944 027 A (FRANCOTYP POSTALIA AG) 22 September 1999 (1999-09-22) claim 1; figure 1	5
Y	US 5 966 714 A (TSO MICHAEL MAN-HAK ET AL) 12 October 1999 (1999-10-12) claim 1; figure 1A	10
	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents:

\*A\* document defining the general state of the art which is not considered to be of particular relevance

\*E\* earlier document but published on or after the international filing date

\*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

\*O\* document referring to an oral disclosure, use, exhibition or other means

\*P\* document published prior to the international filing date but later than the priority date claimed

\*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

\*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

\*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

\*G\* document member of the same patent family

Date of the actual completion of the international search

21 February 2001

Date of mailing of the international search report

28/02/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax (+31-70) 340-3016

Authorized officer

Kirsten, K

# INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 00/41322

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 754 306 A (LYNCH-FRESHNER LAWRENCE A ET AL) 19 May 1998 (1998-05-19) column 17, line 27 - line 33; claim 1; figure 3	13,25
A	WO 98 57302 A (RAHRIG JOHN G ;RILEY DAVID W (US); GRAVELL LINDA V (US); PINTSOV L) 17 December 1998 (1998-12-17) claim 1; figure 5	1-48
A	WO 99 48054 A (SCHWARTZ ROBERT G ;BROOKNER GEORGE (US); ASCOM HASLER MAILING SYS) 23 September 1999 (1999-09-23) claim 1; figure 1A	1-48
A	EP 0 854 448 A (PITNEY BOWES) 22 July 1998 (1998-07-22) claim 1; figure 5A	1-48

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International Application No

PCT/US 00/41322

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0663652 A	19-07-1995	US 5454038 A	26-09-1995
		CA 2137403 A	07-06-1995
		EP 0829824 A	18-03-1998
		US 5682429 A	28-10-1997
		US 5781634 A	14-07-1998
		US 5953427 A	14-09-1999
EP 0944027 A	22-09-1999	DE 19812903 A	23-09-1999
US 5966714 A	12-10-1999	US 5706509 A	06-01-1998
		AU 2219397 A	19-11-1997
		WO 9741520 A	06-11-1997
US 5754306 A	19-05-1998	US 6147773 A	14-11-2000
WO 9857302 A	17-12-1998	AU 7961998 A	30-12-1998
		AU 7963898 A	30-12-1998
		AU 8255898 A	30-12-1998
		AU 8256698 A	30-12-1998
		AU 8567098 A	30-12-1998
		AU 8567298 A	30-12-1998
		BR 9805995 A	31-08-1999
		BR 9806225 A	21-03-2000
		CN 1234890 T	10-11-1999
		CN 1234891 T	10-11-1999
		EP 0925558 A	30-06-1999
		EP 0931299 A	28-07-1999
		EP 0925663 A	30-06-1999
		EP 0920679 A	09-06-1999
		EP 0966728 A	29-12-1999
		EP 0960394 A	01-12-1999
		WO 9857303 A	17-12-1998
		WO 9857460 A	17-12-1998
		WO 9857304 A	17-12-1998
		WO 9857305 A	17-12-1998
		WO 9857306 A	17-12-1998
WO 9948054 A	23-09-1999	EP 1062638 A	27-12-2000
EP 0854448 A	22-07-1998	US 5873073 A	16-02-1999
		CA 2225693 A	24-06-1998

**THIS PAGE BLANK (USPTO)**